~~Cooperative Agreement~~
~~University of Montana~~
Thesis/
Reports
Zuuring, H.

FINAL REPORT FOR RESEARCH AGREEMENT
#RMRS-99544-RJVA
"Enhancing MAGIS Interfaces with GIS and SIMPPLLE"
THE UNIVERSITY OF MONTANA
FS Contact: Mr. Greg Jones
CoOp Contact: Dr. Hans Zuuring

Final Report

For

"Enhancing MAGIS Interfaces with GIS and SIMPPLLE" Phase I (RMRS-99544-RJVA)

Prepared
By

Dr. Hans Zuuring
School of Forestry
The University of Montana
Missoula, MT 59812
(406-243-6456)

July 14, 2000

INTRODUCTION

For a number of years researchers at the School of Forestry, The University of Montana (UM), USDA Forest Service - Intermountain Research Station (INT), and Montana Department of Natural Resources and Conservation (DNRC)- Forestry Division have been developing computer software for planning land management and transportation-related activities on a geographic and temporal basis in the presence of multiple and sometimes conflicting objectives. This computer software was named **MAGIS** that is an acronym for **M**ulti-Resource **A**nalysis and **G**eographic **I**nformation **S**ystem. This system utilizes graphical and tabular data from an external GIS, together with numeric data entered via a series of user input screens. Once the necessary data have been entered the user can build planning scenarios through a SETUP procedure, obtain a solution to the problem and then display the results in tabular or graphical form.

An alternative approach to projecting vegetation over time was implemented so that MAGIS would be more compatible with ecosystem management at landscape scales. This new approach involved the use of vegetative states that move along certain pathways over time in the presence of natural processes. The standard method of vegetation projection utilizes timber strata where DBH, height, basal area, and volume characterize trees. Using an optional tree growth simulator like SPS or PROGNOSIS (FVS) the above attributes are modified over time in the presence or absence of management activities. This capability was added to MAGIS as well as further enhancing, testing and debugging of this modeling system by drawing up an open-ended research joint venture agreement (INT 96038-RJVA) between the USDA Forest Service - INT Research Station and The University of Montana. Phase I of this agreement was completed on March 31, 1997. Phase II started on February 18, 1997 and was completed on March 31, 1998. Phase III was started on April 15, 1998 and completed on May 31, 1999.

A new open-ended BEMR project titled "Enhancing MAGIS Interfaces with GIS and SIMPPLLE" was started between the USDA Forest Service - Rocky Mountain Research Station and the University of Montana on May 5, 1999 and completed on April 15, 2000. The main purpose of this new research project was to streamline data sharing between MAGIS and SIMPPLLE and make MAGIS more user-friendly. It was anticipated that this research would occur over a two-year period with two phases.

OBJECTIVES

A. This agreement continues MAGIS-related research and development efforts initiated under Cooperative Research and Development Agreement INT 92655-CRADA, "MAGIS Agreement" which expired on June 30, 1993, Research Joint Venture Agreement INT 94979-RVJA, "Enhance MAGIS" which ended September 30, 1996 and Research Joint Venture Agreement INT 96038-RVJA, "Vegetative Pathways in MAGIS, Phases II & III" which ended May 31, 1999.

B. This open-ended agreement provides for formal interaction on the following Phase I specific objectives within the overall, joint research and development project:

1) Automate the process of bringing existing ArcView shapefiles and Arc/Info coverages and associated attributes into MAGIS.

2) Develop an intuitive interface that permits the user to interact easily with MAGIS.

3) Complete the conversion of MAGIS code from FoxPro v2.6 to Visual FoxPro v5.0.

4) Improve transfer of data between MAGIS and SIMPPLLE.

5) Modify MAGIS so as to permit long names with associated descriptions for any list that now employs cryptic 8-character names.

6) Complete and update the hardcopy version of the MAGIS User's Guide.

7) Create a web-based version of the MAGIS User's Guide.

8) Serve as a liaison between the Bitterroot Ecosystem Management Research Project and the Bitterroot Water Forum for discussions about collaborative planning across multiple ownerships.

METHODS

Objective 1) will be met partly by awarding a contract to a private consultant after detailed contract specifications have been written. These specifications are presented as a set of notes located in Appendix 1. Progress will be monitored and two research associates already working on this project will make further modifications.

Objective 2) will be met by developing detailed contract specifications so that a private consultant can carry out the work. We will identify mechanisms, screens and graphical interfaces that would be modified or newly coded so as to reduce data entry time.

Objective 3) will be met by writing computer-programming code that will fix old and new bugs. The latter has come about as the result of incorporating new code written by private consultants. In addition the functionality of certain routines will be enhanced and opportunities will be identified where code can be rewritten so as to be more efficient in terms of execution speed.

Objective 4) will involve the creation of conversion routines and crosswalk tables so that data can be readily transferred between MAGIS and SIMPPLLE.

Objective 5) calls for the relaxation of the number of characters for names associated with management relations, costs, outputs and activities from 8 to 25.

Objective 6) will be met by reviewing the content and organization of existing MAGIS documentation and writing three documents namely:

a) "Getting Started" booklet
b) A tutorial document
c) A comprehensive reference guide.

Objective 7) requires the creation of web-based equivalents of the above three documents.

Objective 8) entails attending a number of meetings with the Bitterroot Water Forum to provide assistance and advice to owners engaged in ecosystem modeling and landscape management planning.

RESULTS and DISCUSSION

Objective 1)
MAGIS is a spatially explicit modeling system dependent on GIS spatial data and associated non-spatial attributes that are used to determine costs and calculate outcomes (and therefore influence which projects the optimizer chooses).

MAGIS models use two types of GIS coverages- a polygon coverage and a network (an 'arc' coverage). Users can have both a stand polygon AND a treatment unit polygon coverage, or can work with a single stand coverage, in which case the stand coverage 'becomes' the treatment units. The process of converting an ARCVIEW coverage to MAGIS data includes building numerous related tables specific to stands, treatment units, and network road segments (links), and also creating a DLG (digital line graph) file. This latter file is used by the internal MAGIS graphical interfaces to draw the same polygons and roads that are displayed in ARCVIEW.

Prior to the implementation of this objective, GIS data had to be converted (usually piecemeal) into a tabular form compatible with MAGIS dbase (FoxPro) tables; a very time-consuming process. The reasoning behind this objective is to allow the user to do all the data preparation for MAGIS modeling in ARCINFO or ARCVIEW, rather than doing some in GIS and some more in FoxPro (or other database). An interface in MAGIS connects to ARCVIEW and brings the data into MAGIS, does all the database work, and puts the data in the proper form. This streamlines much database processing and creates an environment that allows the data processing to be done in a way familiar to MAGIS users.

A series of screens was designed in FoxPro that call ARCVIEW, run AVENUE scripts, and convert a set of ARCVIEW coverages (with their associated attributes) into the tables that are used in a MAGIS model. A short-term contract was awarded to a GIS analyst

(Judy Troutwine) to write ARCVIEW Avenue scripts and FoxPro database code to accomplish the various tasks of converting data. J. Greg Jones, Janet Sullivan and Kurt Krueger did the design and monitoring of this contract. FoxPro screens were initiated by Judy Troutwine, then incorporated into MAGIS by Kurt Krueger and Janet Sullivan.

Current Status: The screens and the ARCVIEW scripts (with an exception noted below) have been written and incorporated into the MAGIS application via the menu system. The import process needs to be further automated—it was added 'in front' of the current system, which still has the user run an 'import' process to convert tabular data having English labels to tabular data incorporating internal codes (which allows MAGIS to run much faster.) To make the GIS import process truly seamless and easy this conversion process should be called automatically from the GIS code, rather than the current method where a user has to deal with a series of menu choices and screens.

The DLG-writer script is not yet functional. It turns out that there was no way to make ARCVIEW write the DLG file—our programmer has to write a data processing script that does the same job as this function in ARCINFO. This function is proprietary and could not be directly incorporated into the script—though the data file format is NOT proprietary.

Objective 2)
Much of the time and resources allocated for this objective were channeled into Objective 1), since that objective turned out to be somewhat more complicated and difficult than anticipated. Some screens and menus were evaluated for improvements that could be made.

Current Status: Existing screens and menus were improved so that a more logical pathway was followed in the model-building process. For example, MAGIS incorporates a rule-based 'resource project generator' that assigns management options to treatment units based on a user-defined set of rules. The screens to build the rules for allocating treatment options to polygons were consolidated with the screen for defining the treatments (called 'management regimes' in MAGIS). In this way the RULES for assigning a management regime as an OPTION to a polygon are now a PART of the definition of the management regime itself. This makes the number of menu items and screens to visit smaller, and seems to be a more logical way of setting up model data. Pathway rules for states were also improved by not only applying rules to the beginning state of treatment units but allowing treatment units to grow into states over time and then passing the RULES test.

Related screens, in which the treatment units are processed to either have options assigned to them by MAGIS (using the rules mentioned above) or by the user (coming from the GIS data) were re-designed to have a more logical flow of work. A new graphical utility was incorporated for establishing logging-method/loading node defaults

for the rule-based resource-project generator. Kurt Krueger and Janet Sullivan did this work.

Objective 3)
It was satisfied by fixed the following screens and code associated with them:

1. **Easy Conversion**
   A) Screens: 10, 18, 23, 40, 43pb, 55, 9a_2, def_46b, imp#ok, imp_ok, imp_sol, mid_sol, pbid_imp, regen_ta, reset_55.
   B) Program code: 3, adjancy, bld_1415, combine, corr_55, corridor, rd_trig, security, unrd_bat.

2. **Functional/speed conversion**: 18 (and related tables - 46, import, bldrespro). Due to the fact that we are now dealing with models that have from 1,500 to 10,000 treatment units the building of pathway resource projects in the PreMatrix process has become a bottleneck. Using the FoxPro Coverage Profiler the code associated with building tables _50a, _50b, and _50c was identified as being slow when solving equations. Therefore this code will be modified to improve performance.

3. **Select/deselect screens**: 48a, 18_neigh, 43pb.

4. **Bug fixes**: _47, bld_path, 6a2, 9a, 12, 16, 19, 20a, 20b, 31, 32, 43pd_add, 43padd2, 43padd3, pre_mat, setup_imp, setupl, side_c, solve, specs.

Objective 4)
MAGIS is a spatially explicit planning model that optimizes a schedule of treatments, calculates outcomes (outputs) and economic considerations. It runs in Windows on a PC. SIMPPLLE is a spatially explicit, stochastic model that predicts the patterns of vegetation growth and natural processes on a landscape. It runs in LISP (soon to be JAVA) on the UNIX platform. The modeling strategy using both SIMPPLLE and MAGIS consists of the following:

1) SIMPPLLE is run to make predictions about vegetative growth and natural processes—running a number of simulations from the same 'starting point' (20 to 100 runs) generates information about general trends and specifically predicts the likelihood that a given polygon will experience some natural process that affects its health (processes include bark beetle, root disease, mountain pine beetle, and three intensities of wildfire.)
2) Construct a 'Risk Index' from these predictions for the polygon.
3) Run MAGIS with these risk indices so that the optimizer can choose a course of treatments that best reduces this risk index yet are economically feasible, and do not adversely affect forest health in other ways (excess sediment production for example.)
4) Feed back the selection of treatments to SIMPPLLE to determine its effects on the predicted natural processes.

Note: The risk index for each polygon is provided from SIMPPLLE as a text file of attributes, which is easily added to MAGIS as a database table.

A MAGIS scenario-solution consists of a schedule of treatments (each treatment unit has an assigned management regime, or no action) as a database table. This table has to be converted to a text file that tells SIMPPLLE which treatments to apply to which treatment unit (and when) the table has to be converted to a text file with a specific format. The requirements for the objective at this stage are a method to take multiple scenarios from MAGIS and convert them to named text files in the proper format. This process requires a crosswalk table consisting of management regime names as they occur in MAGIS and the SIMPPLLE-equivalent management activities. A MAGIS management regime might consist of several activities at specified intervals, for example, CT_CCR40 is a commercial thin followed by a clear-cut with reserves 40 years later. This crosswalks to two SIMPPLLE activities in two different periods.

The results of the SIMPPLLE run (consisting of the treatment units and their vegetative state of growth and the processes that occur over time) are output as separate text files indicating, for each polygon, for each period, the treatments that are applied or (in the other file) the processes that occur. The combination of treatment and process for the entire five periods has to be converted to a MAGIS management regime, built into MAGIS and then the outcomes calculated for each scenario (and each simulation run from SIMPPLLE.) Currently, the MAGIS model and LP Matrix have to be built for each set of simulations. Then a calculation program is run (from the same utility) to extract the outcomes for each scenario-simulation (without having to re-enter the scenario for each one) and output the information to a spreadsheet for analysis.

Current Status: The crosswalk tables have been created, and a utility program that runs from FoxPro has been programmed (by Janet Sullivan) that allows the user to either enter one MAGIS solution or SIMPPLLE simulation at a time and get the desired file, or to run a batch process on a large number of MAGIS scenarios or SIMPPLLE simulations at one time, placing the resultant files in a single folder for easy access. Future improvements for this utility might include running it directly from SIMPPLE, and/or running a calculation routine without having to build the MAGIS model and LP Matrix each time.

Objective 5)

Current Status: Some testing of the feasibility of increasing field lengths has been done; probably most screens can be modified to accept longer names and characters, but this has not yet been accomplished. Effort has been concentrated on conversion of screens to Visual FoxPro. Changes to tables will require some additions to the table-checking/modification program that runs when a new model is opened in the DBF directory. The four screens that manage ZIP files (the data set for a given model is stored as a zip file consisting of all the tables specific to the model) have been completely converted to Visual FoxPro v5.0. An updated version of PKZIP (and PKUNZIP) which is now one program (PKZIP25 - a file compression utility) has also been incorporated

into MAGIS. This updated version allows windows-length file names (instead of taking long file names and converting them to DOS names with a ~ for the extra characters.) In addition, the user description of the data set is no longer limited to 20 characters, and can be updated with each save of the model.

Objective 6)
Progress was made on MAGIS documentation but a lot still needs to be done. Considerable time was spent on semantics and the manner in which the MAGIS story should be told and what type of graphics should be included. About 30 pages of notes were written. A 15-page document containing MAGIS training exercises was written for the USFS Regional Training Academy held on the U of M campus on Friday March 24, 2000. In addition a 30-page MAGIS tutorial - Upper Belt model document was also written. Therefore, Objective 6) was partially satisfied.

Objective 7)
It was not satisfied because the MAGIS User's Guide was not finished although some time was spent making revisions to HTML documents.

Objective 8)
Between March 1999 and December 1999 Mike Sweet participated in over a dozen meetings with the Water Forum. From October 1999 to January 2000 he worked with the BEMRP Human Dimensions Group to formulate a Water Forum related proposal for funding by BEMPR in FY2000. This activity satisfied Objective 8).

Appendix 1.

NOTES FOR STREAMLINING THE PROCESS OF BRINGING EXISTING ARCVIEW COVERAGES AND ASSOCIATED ATTRIBUTES INTO MAGIS
(Oct 1, 1999)

I. Stand Data

The MAGIS data tables that are populated with stand data are:

- PATH_MST.DBF (for the Pathway option for projecting vegetation) or STD_MSTR.DBF (for the standard method for projecting vegetation)
- STD_ATTR.DBF

A FoxPro stand data form is needed (accessed from the MAGIS menu) that contains a process to populate these data tables.

- Need to ask the user for the location and name of the coverage containing the stand data. This will likely reside on a Unix machine, networked to the PC running MAGIS. But it could reside on the PC, because the Forest Service now has a site license for PC ArcView.
  - ☞ Need to identify which attribute is the stand id
- Notes related to STD_ATTR.DBF:
  - ☞ We envision a portion of the stand form that does the following:
    - -- Contains a "candidate" list that includes all the attributes in the GIS stand coverage that meet the "candidate" requirements listed in Section V.
      - * Double clicking on an attribute moves it from the "candidate" list to the "selected" list
    - -- Contains a "selected" list that includes all the attributes from the GIS coverage that are to be loaded into STD_ATTR.DBF
    - -- Contains an option for displaying a list of the "required" attributes.
    - -- Contains an option for displaying a list of the "BAD" attributes along with each of the problems associated with the bad attribute (Info for "BAD" attributes comes from Section V). Also provide an option of printing this information.
    - -- Contains a button for processing the attributes on the "selected" list.
  - ☞ Build the list of the "candidate" and "BAD" attributes from the attributes found in the GIS data
    - -- Build the list of GIS attributes to send to the process in Section V. This includes all attributes in the GIS stand coverage except: Stand Id, the attributes going into PATH_MST.DBF (HBTY_GRP, DOM_SP, SZ_CLASS, DENSITY, TIME_INC), and the attributes going into STD_MSTR.DBF (For_ty, Stock, Prod, Seral_ss, Hab_ty, Age, Yr_into_ss, Vol, Ba, Dbh, Ht).

-- Call steps in Section V to provide: (1) list of "candidate" attributes, (2) list of "BAD" attributes and the problem(s) of each.

☞ Build the list of "required" attributes. This will use the process Kurt has for building a list of stand attributes needed for the outputs and costs present in the MAGIS model.

☞ Build the default list of "selected" attributes

-- Move each GIS attribute whose name matches a "required" attribute from the "candidate" list to the "selected" list.

☞ Button for processing the attributes on the "selected" list.

-- Check to see if all "required" attributes have been included in the "selected" list. If not, write message saying: "The following "required" attributes are missing from the "selected" list: ____, _____"

-- If all "required" attributes are on the "selected" list, process the attributes as follows:

* For each "selected" attribute, add any necessary information to _12a.dbf. To do this, check _12a.dbf to determine if the attribute is present.

(1) If present ==> check the map element type for that attribute in _12a.dbf

(a) If map element type = 'S', or 'LS' ==> no change is needed in _12a.dbf

(b) If map element type = 'L' ==> change to 'LS'

(c) Otherwise (map element type = 'O', 'C', 'LC')==> cannot bring that attribute into MAGIS with its current name. Must rename.

(2) If NOT present

(a) Write new record to _12a.dbf for this attribute:

Attrib Type = 'A' (for attribute)
Internal ID = Use same logic as screen set for _12a to assign a number
Attrib/Zone Name: Name in GIS
Numeric Type: This is determined in building the "candidate" list (Section V)

Numeric Type = 'D'
Continuous Type = 'C'

Map Element Type = 'S'
Attribute Source = leave blank
Attribute Classes Defined?
'T' if Numeric Type = 'D'
'F' if Numeric Type = 'C'

(b) For Numeric Type = 'D' only, write new record to _12b.dbf for each class found in GIS data for this attribute:

Internal ID = Match number entered in _12a for this attribute

Discrete Class Name = Class name in GIS
Continuous Upper Limit = (leave blank)
(Note, somehow we need to ensure that the user goes to the screen set for attributes and completes this information for attributes that are continuous.)
Internal Code = Use same logic as screen set for _12b

* Copy the "selected" attribute information for each polygon to file STD_ATTR.DBF (see attached documentation and example for format of this file)

- Notes related to PATH_MST.DBF (This file is built if table _32.strd_path = 2, indicating the veg projection type is pathway)
  - The attributes needed are: HBTY_GRP, DOM_SP, SZ_CLASS, DENSITY, TIME_INC
  - We envision a process that looks for each of these attributes in the GIS data, when found checks the data, then if all is ok builds PATH_MST.DBF. If problems are found a report is written to the screen with an option to print.
    -- Find each of the attribute names in the GIS data. If one of the "required" attributes is not found, write the following error message: "Required Attribute XXXXXXX was not found in the GIS data." (where XXXXXXX represents the name of the missing attribute.)
    -- For each attribute check to determine if each of the GIS values for that attribute is in MAGIS table _12b.dbf. If one or more GIS values are not present in _12b.dbf write the following error message: "The following classes for XXXXXXX are present in the GIS data, but are not defined in MAGIS: ___, ____, etc."
    -- If no errors are found in the PATH_MST.DBF steps above, write the information from GIS for each polygon to file PATH_MST.DBF (see attached documentation and example for format of this file)

- **Notes related to STD_MSTR.DBF** (This file is built if table _32.strd_path = 1, indicating the veg projection type is standard)
  - The attributes needed include both discrete attributes and continuous (numeric) attributes
  - The discrete attributes needed are: For_ty, Stock, Prod, Seral_ss, Hab_ty
    -- Find each of the attribute names in the GIS data. If one of the "required" attributes is not found, write the following error message: "Required Attribute XXXXXXX was not found in the GIS data." (where XXXXXXX represents the name of the missing attribute.)
    -- For each attribute check to determine if each of the GIS values for that attribute is in MAGIS table _12b.dbf. If one or more GIS values are not present in _12b.dbf write the following error message: "The following classes for XXXXXXX are present in the GIS data, but are not defined in MAGIS: ___, ____, etc."

- ☞ The continuous attributes needed are: Age, Yr_into_ss, Vol, Ba, Dbh, Ht
  - -- Find each of the attribute names in the GIS data. If one of the "required" attributes is not found, write the following error message: "Required Attribute XXXXXXX was not found in the GIS data." (where XXXXXXX represents the name of the missing attribute.)
- ☞ If no errors are found in the STD_MSTR.DBF steps above, write the information from GIS for each polygon to file STD_MSTR.DBF (see attached documentation and example for format of this file)

II. Treatment Unit Data (alias, Cutting Unit)

The MAGIS data tables that are populated with treatment unit data are:

- ♦ CU_MSTR.DBF
- ♦ CU_ZONE.DBF
- ♦ CUTPROP.DBF
- ♦ CU_MGNT.DBF (Data for this file, however, will not be obtained from GIS)

A FoxPro treatment unit polygon form to be run from the MAGIS menu with a process to populate these data tables is needed.

- ♦ Need to ask the user for the location and name of the coverage containing the treatment unit data. This will likely reside on a Unix machine, networked to the PC running MAGIS. But it could reside on the PC, because the Forest Service now has a site license for PC ArcView.
  - ☞ Need to identify which attribute is the Treatment Unit Id
- ♦ **Notes related to CU_ZONE.DBF:**
  - ☞ We envision a portion of the treatment unit form that does the following:
    - -- Contains a "candidate" pick list that includes all the attributes in the GIS treatment unit coverage that meet the "candidate" requirements listed in Section V.
      - * Double clicking on an attribute moves it from the "candidate" list to the "selected" list
    - -- Contains a "selected" list that includes all the attributes from the GIS coverage that are to be loaded into CU_ZONE.DBF
    - -- Contains an option for displaying a list of the "required" attributes.
    - -- Contains an option for displaying a list of the "BAD" list of attributes along with each of the problems associated with the bad attribute (Info for "BAD" attributes comes from Section V). Also provide an option of printing this information.
    - -- Contains a button for processing the attributes on the "selected" list.
  - ☞ Build the list of potential attributes from the attributes found in the GIS data
    - -- Build the list of GIS attributes to send to the process in Section V. This includes all attributes in the GIS stand coverage except: the Treatment Unit Id.

-- Call steps in Section V to provide: (1) list of "candidate" attributes, (2) list of "Bad" attributes the problem(s) of each.

☞ Build the list of "required" attributes. This is the list of treatment unit zone attributes needed for the Management Regime Allocator present in the MAGIS model.

☞ Build the default list of "selected" attributes

-- Move each GIS attribute whose name matches a "required" attribute from the "candidate" list to the "selected" list.

☞ Button for processing the attributes on the "selected" list.

-- Check to see if all "required" attributes have been included in the "selected" list. If not, write message saying: "The following "required" attributes are missing from the "selected" list: ____, _____"

-- If all "required" attributes are on the "selected" list, process the attributes as follows:

* For each "selected" attribute, add any necessary information to _12a.dbf. To do this, check _12a.dbf to determine if the attribute is present.

(1) If present ==> check the map element type for that attribute in _12a.dbf

(a) If map element type = 'C', or 'LC' ==> no change is needed in _12a.dbf

(b) If map element type = 'L' ==> change to 'LC'

(c) Otherwise (map element type = 'O', 'S', 'LS')==> cannot bring that attribute into MAGIS with its current name. Must rename.

(2) If NOT present

(a) Write new record to _12a.dbf for this attribute:

Attrib Type = 'Z' (for attribute)
Internal ID = Use same logic as screen set for _12a to assign a number
Attrib/Zone Name: Name in GIS
Numeric Type = 'D'
Map Element Type = 'C'
Attribute Source = leave blank
Attribute Classes Defined? = 'T'

(b) Write new record to _12b.dbf for each class found in GIS data for this attribute:

Internal ID = Match number entered in _12a for this attribute
Discrete Class Name = Class name in GIS
Continuous Upper Limit = (leave blank)
Internal Code = Use same logic as screen set for _12b

* Copies the information for each polygon to file CU_ZONE.DBF (see attached documentation and example for format of this file)

- **Notes related to CU_MSTR.DBF**
  - The attributes needed are: PAS_ACT_ID, YR_SIN_ACT
  - The attributes needed include both discrete attributes and continuous (numeric) attributes
  - The discrete attribute needed is: PAS_ACT_ID
    - -- Find this attribute name in the GIS data. If it is not found, write the following warning message: "Attribute PAS_ACT_ID was not found in the GIS data."
    - -- For each attribute check to determine if each of the GIS values for PAS_ACT_ID is in MAGIS table _37.dbf. If one or more GIS values are not present in _37.dbf write the following error message: "The following classes for PAS_ACT_ID are present in the GIS data, but are not defined in MAGIS: ___, ____, etc."
  - The continuous attribute needed is: YR_SIN_ACT
    - -- Find this attribute name in the GIS data. If it is not found, write the following warning message: "Attribute YR_SIN_ACT was not found in the GIS data."
  - If no errors are found in the CU_MSTR.DBF steps above (warnings ok), write the information from GIS for each polygon to file CU_MSTR.DBF (see attached documentation and example for format of this file). **Slope is included as an attribute in CU_MSTR.DBF, but it is no longer used. Place "-99" in each record for slope.**
- **Notes related to CUTPROP.DBF**
  - This file lists for each treatment unit, the number of acres of each stand included in that unit
  - If the stand polygons are used as the treatment unit polygons (ie. the same GIS coverage is used for both), then the only information needed to build CUTPROP.DBF is the number of acres in each polygon
  - If the treatment unit polygons do not match the stand polygon ===> we need a procedure or GIS instructions to develop this information.

III. Road Link Data

The MAGIS data tables that are populated with road link data are:

- LNK_MSTR.DBF
- LNK_ATZN.DBF

A FoxPro road link form is needed (accessed from the MAGIS menu) that contains a process to populate these data tables.

- Need to ask the user for the location and name of the coverage containing the road data. This will likely reside on a Unix machine, networked to the PC running MAGIS. But it could reside on the PC, because the Forest Service now has a site license for PC ArcView.

- ☞ Need to identify which attributes are the From_node and To_node

- **Notes related to LNK_ATZN.DBF:**
  - ☞ We envision a portion of the road form that does the following:
    -- Contains a "candidate" pick list that includes all the attributes in the GIS road coverage that meet the "candidate" requirements listed in Section V.
      * Double clicking on an attribute moves it from the "candidate" list to the "selected" list
    -- Contains a "selected" list that includes all the attributes from the GIS coverage that are to be loaded into LNK_ATZN.DBF
    -- Contains an option for displaying a list of the "required" attributes.
    -- Contains an option for displaying a list of the "BAD" attributes along with each of the problems associated with the bad attribute (Info for "BAD" attributes comes from Section V). Also provide an option of printing this information.
    -- Contains a button for processing the attributes on the "selected" list.
  - ☞ Build the list of potential attributes from the attributes found in the GIS data
    -- Build the list of GIS attributes to send to the process in Section V. This includes all attributes in the GIS road coverage except: From_node, To_node, and the attributes going into LNK_MSTR (HORZ_LEN, CUR_STATUS, RD_OPTION, ALIGMNT, SURF_TY, #_LANES, %_GRADE, TERR_N, RD_MTCE, SPEED, CONST_YR).
    -- Call steps in Section V to provide: (1) list of "candidate" attributes, (2) list of "Bad" attributes and the problem(s) of each.
  - ☞ Build the list of "required" attributes. This will use the process Kurt has for building a list of stand attributes needed for the outputs and costs present in the MAGIS model.
  - ☞ Build the default list of "selected" attributes
    -- Move each GIS attribute whose name matches a "required" attribute from the "candidate" list to the "selected" list.
  - ☞ Button for processing the attributes on the "selected" list.
    -- Check to see if all "required" attributes have been included in the "selected" list. If not, write message saying: "The following "required" attributes are missing from the "selected" list: ____, _____"
    -- If all "required" attributes are on the "selected" list, process the attributes as follows:
      * For each "selected" attribute, add any necessary information to _12a.dbf. To do this, check _12a.dbf to determine if the attribute is present.
        (1) If present ==> check the map element type for that attribute in _12a.dbf
          (a) If map element type = 'L', 'LC' or 'LS' ==> no change is needed in _12a.dbf
          (b) If map element type = 'S' ==> change to 'LS'
          (c) If map element type = 'C' ==> change to 'LC'

(d) Otherwise (map element type = 'O') ==> cannot bring that attribute into MAGIS with its current name. Must rename.

(2) If NOT present

(a) Write new record to _12a.dbf for this attribute:

Attrib Type = 'A' (for attribute) or 'Z' (zone locator)
If the attribute is used for an output and therefore on Kurt's "required" list use 'A'. Otherwise, we need to ask the user.

Internal ID = Use same logic as screen set for _12a to assign a number

Attrib/Zone Name: Name in GIS

Numeric Type: This is determined in building the "candidate" list (Section V)
Numeric Type = 'D'
Continuous Type = 'C'.

Map Element Type = 'L'

Attribute Source = leave blank

Attribute Classes Defined?
'T' if Numeric Type = 'D'
'F' if Numeric Type = 'C'

(b) For Numeric Type = 'D' only, write new record to _12b.dbf for each class found in GIS data for this attribute:

Internal ID = Match number entered in _12a for this attribute

Discrete Class Name = Class name in GIS

Continuous Upper Limit = (leave blank)

(Note, somehow we need to ensure that the user goes to the screen set for attributes and completes this information for attributes that are continuous.)

Internal Code = Use same logic as screen set for _12b

* Copy the "selected" GIS attribute information for each road segment to file LNK_ATZN.DBF (see attached documentation and example for format of this file)

- **Notes related to LNK_MSTR.DBF:**
  - Three attributes are always "required", then others may or may not be "required", depending on the outputs to be calculated, whether maintenance cost is computed, and whether the travel time method is used to compute haul cost. The steps below determine what is "required" (build a "required" list), check the list, and then write the LNK_MSTR.DBF file.
  - Build the "required" attributes list:
    -- The following attributes are always "required": HORZ_LEN, CUR_STATUS, RD_OPTION.

-- Add any LNK_MSTR attributes "required" for outputs produced from either road projects or traffic. To do this check _21.dbf for any active outputs with source type = 'R' or 'T'. For these determine if any of the following attributes are used. (Note the names for the internal codes are in _12a.dbf) If so, add that attribute to the "required" List:

_12a Attribute Name
ALIGMNT
SURF_TY
#_LANES
%_GRADE
TERR_N

-- Check to determine if road maintenance cost option is used, and if yes, add the "required" attributes. To do this check, in the GIS data look for a field called "RD_MTCE". If RD_MTCE is not not found, we assume that road maintenance cost option is not used. If found, if RD_MTCE contains one or more entries = "T" then add the following to the "required" attribute list: RD_MTCE, SURF_TY, and #_LANES.

-- Add attribute CONST_YR to the "required" list if it is found in the GIS data.

-- Determine if the LNK_MSTR attribute fields for computing traffic cost will be needed:

* On the FoxPro form for collecting info for LNK_MSTR, ask "Will traffic cost be computed?" Provide the following as mutually exclusive choices:
  - ❍ No
  - ❍ Yes, use speed entered by road segment
  - ❍ Yes, use table look-up for travel times
  - ❍ Yes, combination of speed and table lookup for travel times
* Based on the response to the above question, the required field attributes are:
  - ❍ No ==> No attributes required
  - ❍ Yes, use speed entered by road segment
    ==>Required attribute: SPEED
  - ❍ Yes, use table look-up for travel times
    ==>Required attributes: ALIGMNT, SURF_TY, #_LANES %_GRADE
  - ❍ Yes, combination of speed and table lookup
    ==>Required attributes: SPEED, ALIGMNT, SURF_TY, #_LANES %_GRADE

☞ Check the validity of the attributes on the "required" list

-- Find each in the GIS data. Write the following message for any missing attribute: "Required Attribute XXXXXXX was not found in the GIS data." (where XXXXXXX represents the name of the missing attribute.)

-- Check that the numeric attributes HORZ_LEN, CONST_YR, SPEED are each numeric field in the GIS data.
  * If any of these are character fields write the following message to the user: "XXXXXX must be a numeric field."
-- Check that all entries are valid for the character fields. If invalid character strings are found for any of the attributes, write the following message to the user: "The following classes for XXXXXXX are present in the GIS data, but are not defined in MAGIS: ___, ____, etc."
  * CUR_STATUS: valid entries are "C" or "E" or null
  * RD_OPTION: each code in GIS should match a code in table _7.dbf, field Rd_option
  * RD_MTCE: valid entries are "T" or "F" or null
  * %_GRADE: each code in GIS should match a code in table _8.dbf, field Grade. This contains integer values between 21 and -21, so another way to check would be for integer values between (and including) these two numbers.
  * ALIGMNT, SURF_TY, #_LANES, TERR_N: The valid entries are in table _12b.dbf, which is related to _12a.dbf where these names are stored.

☞ Build the LNK_MSTR.DBF file (see attached documentation and example for format of this file)
-- Do this step only if no errors are found in the LNK_MSTR.DBF data
-- For the attributes on the "required" list, transfer the values from GIS to LNK_MSTR.DBF
-- Leave null the attributes not on the "required" list. For the LNK_MSTR.DBF field "Rd_close" use "F" in all cases.

IV. Spatial Data

The files needed by MAGIS are:

- ROAD.DLG (contains road link spatial data)
- POLYGON.DLG (contains treatment unit spatial data)
- GIS_C_ID.DBF (cross-walk table between the labels for treatment units in POLYGON.DLG and the labels used in MAGIS)
- GIS_L_ID.DBF (cross-walk table between the labels for links in ROAD.DLG and the labels used in MAGIS)
- GIS_N_ID.DBF (cross-walk table between the labels for nodes in ROAD.DLG and the labels used in MAGIS)

Currently, ROAD.DLG and POLYGON.DLG must be written out of ARCINFO. If there were a way for these files to be created from ARCVIEW (or created from information from ARCVIEW), that would be desirable.

An automated procedure for creating GIS_C_ID.DBF, GIS_L_ID.DBF, and GIS_N_ID.DBF is needed. We have built these manually to this point.

V. Building the Attribute "candidate" List and List of "Bad" Attributes

- We will store the "BAD" attributes in a temporary table or array that has two fields
  - Field 1: Attribute name
  - Field 2: Problem Statement
    -- The possible problems are listed below
    1. GIS attribute name exceeds 8 characters
    2. One or more GIS class names exceed 8 characters
    3. Attribute name exists in MAGIS, but GIS data contains one or more class names that are not in MAGIS

- The Map Element Type defined in MAGIS for this attribute (XX) is incompatible with the type in the GIS data (YY). (Note, XX represents the Map Element Type defined in MAGIS; and YY represents the map element type being processed -- Stand, Treatment Unit, or Road Segment)
  - Process each attribute sent from the calling routine with the following logic

    - If the GIS attribute is NOT in _12a.dbf ==> add to Attribute "candidate" list if NONE of the following problems are found:
      -- If the GIS attribute is discrete (check for the type of field, if character, then the attribute is discrete)
      1. If the GIS attribute name > 8 characters ==> write attribute to BAD list, record problem statement 1
      2. If one or more class names for that attribute in the GIS data exceed 8 characters ==> write attribute to Bad list, record problem statement 2

      -- If the GIS attribute is continuous (check for the type of field, if numeric, then the attribute is continuous)
      1. If the GIS attribute name > 8 characters ==> write attribute to Bad list, record problem statement 1

    - If the GIS attribute is in _12a.dbf ==> add to Attribute "candidate" list if NONE of the following problems are found:
      -- If the attribute is Numeric Type = 'D' (discrete) and if one or more class names for that attribute in the GIS data exceed 8 characters ==> write attribute to BAD list, record problem statement 2
      -- If the attribute is Numeric Type = 'D' (discrete) and if one or more class names for that attribute in the GIS data do not match a class name for that attribute in _12b.dbf ==> write attribute to BAD list, record problem statement 3
      -- If the calling routine is processing stand data

    * If the Map Element Type = 'C', 'LC', or 'O' ==> write attribute to BAD list, record problem statement 4

-- If the calling routine is processing treatment unit data
    * If the Map Element Type = 'S', 'LS', or 'O' ==> write attribute to BAD list, record problem statement 4

-- If the calling routine is processing road segment data
    * If the Map Element Type = 'O' ==> write attribute to BAD list, record problem statement 4